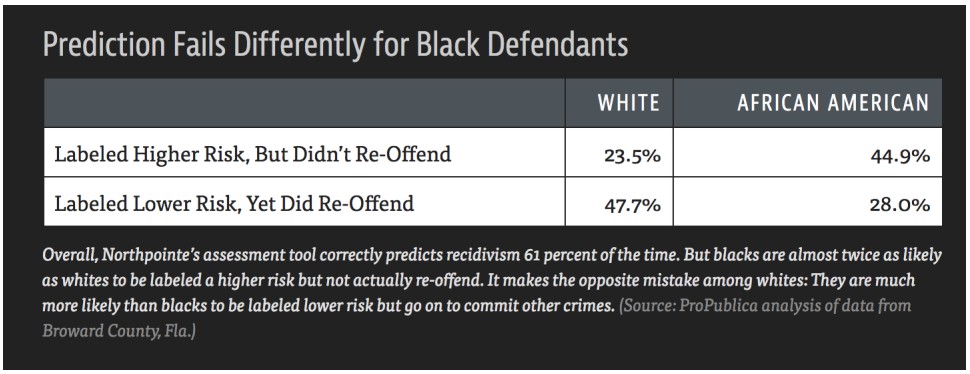
Stats for Clinical Trials, Math 150
Jo Hardin
Survival Analysis, R code

First let's talk about this:
`https://www.propublica.org/article/machine-bias-risk-assessments-in-criminal-sentencing`

> If computers could accurately predict which defendants were likely to commit new crimes, the criminal justice system could be fairer and more selective about who is incarcerated and for how long. The trick, of course, is to make sure the computer gets it right. If its wrong in one direction, a dangerous criminal could go free. If its wrong in another direction, it could result in someone unfairly receiving a harsher sentence or waiting longer for parole than is appropriate.



Prediction Fails Differently for Black Defendants

	WHITE	AFRICAN AMERICAN
Labeled Higher Risk, But Didn't Re-Offend	23.5%	44.9%
Labeled Lower Risk, Yet Did Re-Offend	47.7%	28.0%

Overall, Northpointe's assessment tool correctly predicts recidivism 61 percent of the time. But blacks are almost twice as likely as whites to be labeled a higher risk but not actually re-offend. It makes the opposite mistake among whites: They are much more likely than blacks to be labeled lower risk but go on to commit other crimes. (Source: ProPublica analysis of data from Broward County, Fla.)

Aside: let's say you want a test for HIV. The "best" test might be the one that says *no one* has HIV! Let's say the sensitivity of the test (true positives) is 0.95 and the specificity of the test (true negatives) is 0.98.

	has HIV	doesn't have HIV	
test positive	9,500	19,800	29,300
test negative	500	970,200	970,700
	10,000	990,000	1,000,000

Recidivism in North Carolina

"This data collection examines the relationship between individual characteristics and recidivism for [a cohort] of inmates released from North Carolina prisons [in 1980]. The survey contains questions on the background of the offenders, including their involvement in drugs or alcohol, level of schooling, nature of the crime resulting in the sample conviction, number of prior incarcerations and recidivism following release from the sample incarceration. The data collection also contains information on the length of time until recidivism occurs." `http://www.icpsr.umich.edu/icpsrweb/NACJD/studies/8987?geography=North+Carolina`

Note that the data are not to be redistributed. Additionally, any intentional identification of a research subject or unauthorized disclosure of his or her confidential information violates the promise of confidentiality given to the providers of the information.

There is some literature available here: `http://www.icpsr.umich.edu/icpsrweb/NACJD/studies/8987?geography=North+Carolina`. Including a paper which gives background on survival analysis generally and uses the data as a case study. (Keeping in mind that the paper was written in 1991 when software programs like R were only just starting to become widely available. The authors used FORTRAN to analyze the data.)

Chung, Ching-Fan, Schmidt, Peter, Witte, Ann D. Survival analysis: A survey. *Journal of Quantitative Criminology. 7*, (1), 59-98, 1991.

Apply survival functions in R using recidivism data with the following variables:

timefollow time from release to return to prison or study end

recid 1 if returned to prison; 0 if not

married 1 if married at time of release from prison; 0 if not

age (in months) at time of release

person 1 if conviction was for a crime against a person; 0 if not

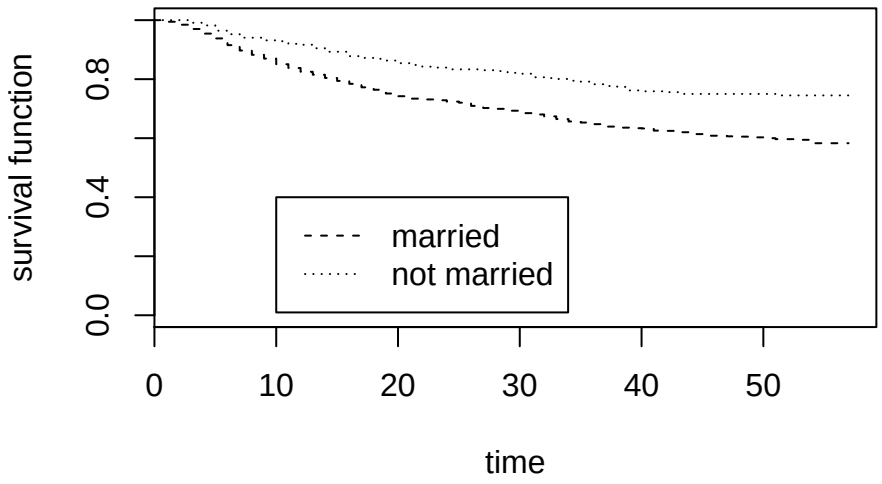
felon 1 if convited for a felony; 0 if not

Importing the data & using the `survival` package

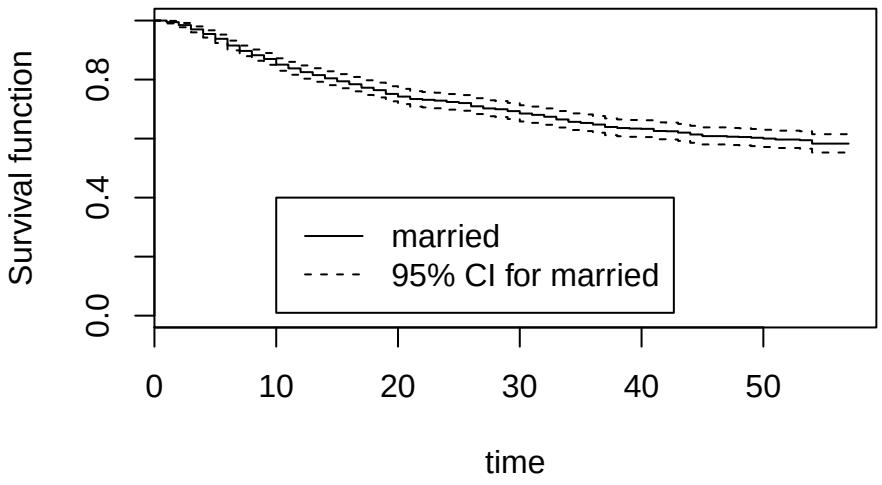
```
library(survival)
recid <- read_csv("~/Dropbox/teaching/math150/recid1980grp1.csv")
recid <- recid %>%
  filter(timefollow > 0)
```

Kaplan-Meier survival curve

```
recid.surv <- survfit(Surv(timefollow,recid) ~ married, data=recid)
plot(recid.surv, lty=2:3,xlab="time", ylab="survival function")
legend(10,.4, c("married", "not married"),lty=2:3)
```

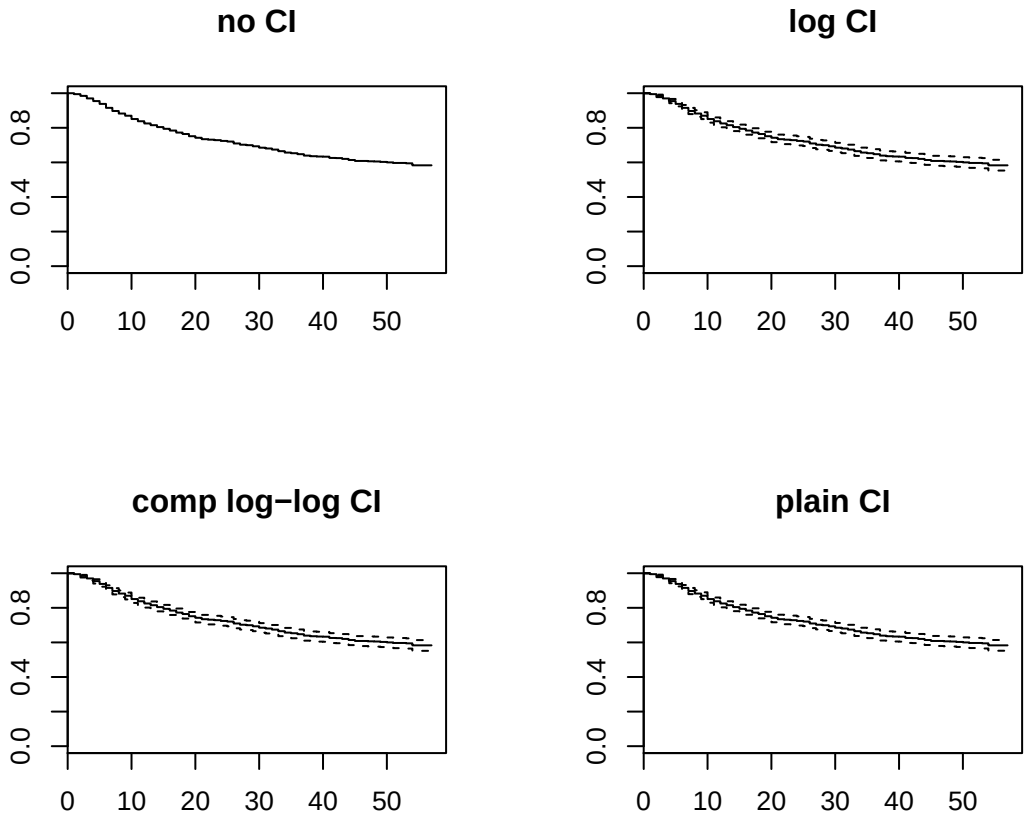



```
plot(recid.surv[1],conf.int = T, xlab="time", ylab="Survival function")
legend(10,.4,c("married", "95% CI for married"), lty=1:2)
```



different options for CI

```
par(mfrow=c(2,2))
plot(survfit(Surv(timefollow,recid) ~ married, data=recid,
             conf.type="none")[1],conf.int=T, main="no CI")
plot(survfit(Surv(timefollow,recid) ~ married, data=recid,
             conf.type="log")[1],conf.int=T, main="log CI")
plot(survfit(Surv(timefollow,recid) ~ married, data=recid,
             conf.type="log-log")[1],conf.int=T, main="comp log-log CI")
plot(survfit(Surv(timefollow,recid) ~ married, data=recid,
             conf.type="plain")[1],conf.int=T, main="plain CI")
```



Log-rank test [rho=0] and the Wilcoxon test [rho=1]

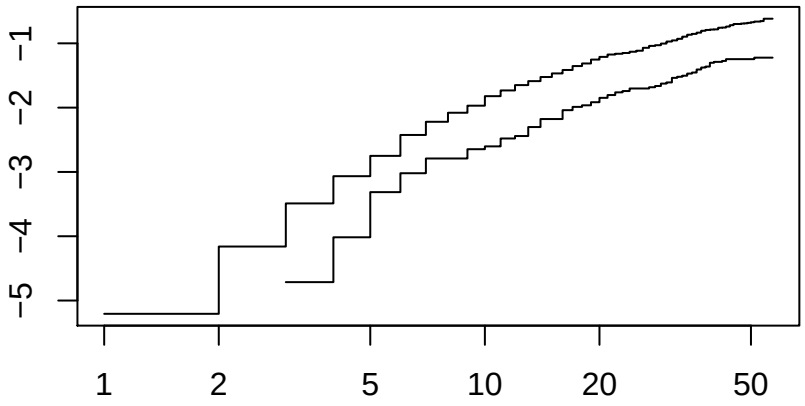
```
survdiff(Surv(timefollow,recid) ~ married, data=recid, rho=0)

## Call:
## survdiff(formula = Surv(timefollow, recid) ~ married, data = recid,
##     rho = 0)
##
##              N Observed Expected (O-E)^2/E (O-E)^2/V
## married=0 1098      444      394      6.27      24.9
## married=1  336       85      135     18.35      24.9
##
##  Chisq= 24.9  on 1 degrees of freedom, p= 6.02e-07

survdiff(Surv(timefollow,recid) ~ married, data=recid, rho=1)

## Call:
## survdiff(formula = Surv(timefollow, recid) ~ married, data = recid,
##     rho = 1)
##
##              N Observed Expected (O-E)^2/E (O-E)^2/V
## married=0 1098    365.4      324      5.23      25.2
## married=1  336     68.7      110     15.43      25.2
##
##  Chisq= 25.2  on 1 degrees of freedom, p= 5.27e-07
```

Cox Proportional Hazards models

```
# Just married
coxph(Surv(timefollow,recid) ~ married, data=recid)

## Call:
## coxph(formula = Surv(timefollow, recid) ~ married, data = recid)
##
##           coef exp(coef) se(coef)     z       p
## married -0.582     0.559    0.118 -4.92 8.8e-07
##
## Likelihood ratio test=27.5  on 1 df, p=1.61e-07
## n= 1434, number of events= 529

coxph(Surv(timefollow,recid) ~ married, data=recid)$loglik

## [1] -3723.696 -3709.967

# married and person
coxph(Surv(timefollow,recid) ~ married + person, data=recid)

## Call:
## coxph(formula = Surv(timefollow, recid) ~ married + person, data = recid)
##
##           coef exp(coef) se(coef)     z       p
## married -0.576     0.562    0.118 -4.86 1.2e-06
## person  -0.391     0.677    0.156 -2.51   0.012
##
## Likelihood ratio test=34.5  on 2 df, p=3.3e-08
## n= 1434, number of events= 529

coxph(Surv(timefollow,recid) ~ married + person, data=recid)$loglik

## [1] -3723.696 -3706.469

# married, person, and felon
coxph(Surv(timefollow,recid) ~ married + person + age, data=recid)

## Call:
## coxph(formula = Surv(timefollow, recid) ~ married + person +
##     age, data = recid)
##
##              coef exp(coef)  se(coef)     z       p
## married -0.460375  0.631047  0.122926 -3.75 0.00018
## person  -0.347061  0.706762  0.156350 -2.22 0.02643
## age     -0.001549  0.998452  0.000443 -3.50 0.00047
##
## Likelihood ratio test=47.8  on 3 df, p=2.33e-10
## n= 1434, number of events= 529

coxph(Surv(timefollow,recid) ~ married + person + age, data=recid)$loglik

## [1] -3723.696 -3699.790
```

Checking proportional hazards with the plot of $\ln(-\ln(S(t)))$

```
plot(survfit(Surv(timefollow,recid) ~ married, data=recid),fun="cloglog")
```



The cox.zph function will test proportionality of all the predictors in the model by creating interactions with time using the transformation of time specified in the transform option. In this example we are testing proportionality by looking at the interactions with log(time). The column rho is the Pearson product-moment correlation between the scaled Schoenfeld residuals and log(time) for each covariate. The last row contains the global test for all the interactions tested at once. A p-value less than 0.05 indicates a violation of the proportionality assumption.

Checking proportional hazards with cox.zph

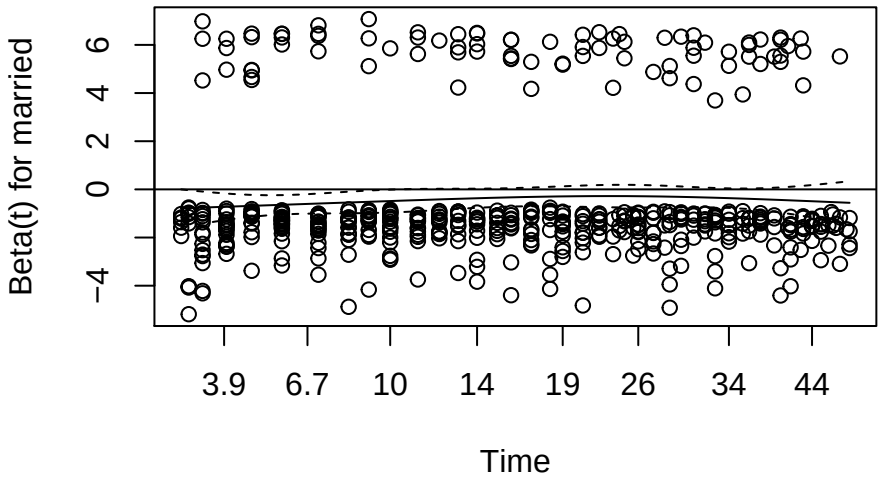
```
cox.zph(coxph(Surv(timefollow,recid) ~ married, data=recid), transform="log")

##           rho chisq     p
## married 0.0365 0.703 0.402

cox.zph(coxph(Surv(timefollow,recid) ~ married + person + age, data=recid))

##             rho chisq     p
## married  0.0385 0.777 0.378
## person   0.0676 2.419 0.120
## age     -0.0549 1.715 0.190
## GLOBAL       NA 4.194 0.241
```

Note the big p-values. We do not reject the null hypothesis, so we conclude that there is no evidence of non-proportional hazards. If for example, the model seemed to be non-proportional on time but proportional on log(time), you might consider transforming the time variable (i.e., taking the natural log) in your original model.

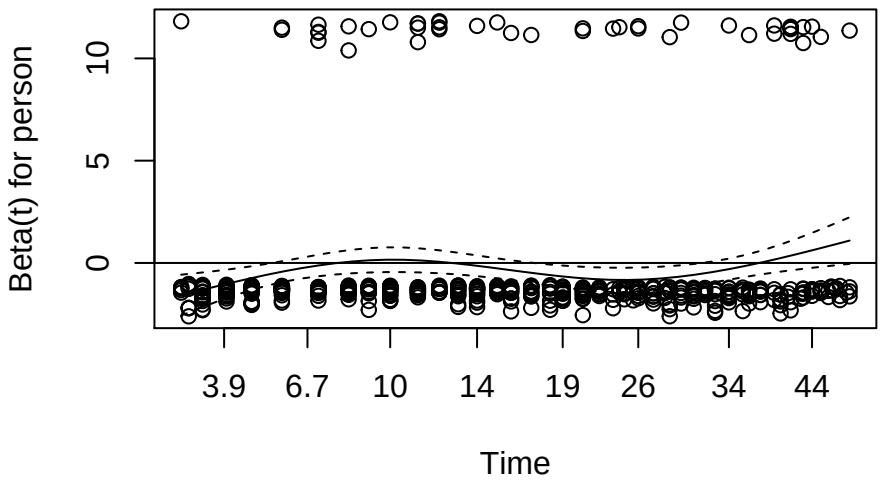
The function cox.zph creates a cox.zph object that contains a list of the scaled Schoenfeld residuals. The ordering of the residuals in the list is the same order as the predictors were entered in the cox model. So,

the first element of the list corresponds to the scaled Schoenfeld residuals for married, the second element corresponds to the scaled Schoenfeld residuals for person, and so forth. The cox.zph object can be used in a plot function. By specifying a particular element of the list it is possible to generate plots of residuals for individual predictors. Leaving out the list number results in plots for all the predictors being generated at one time. In the plots a non-zero slope is evidence against proportionality. The horizontal line at y=0 has been added for reference.

```
plot(cox.zph(coxph(Surv(timefollow,recid) ~ married + person + age, data=recid))[1]); abline(h=0)
```




```
plot(cox.zph(coxph(Surv(timefollow,recid) ~ married + person + age, data=recid))[2]); abline(h=0)
```




```
plot(cox.zph(coxph(Surv(timefollow,recid) ~ married + person + age, data=recid))[3]); abline(h=0)
```